

Freeform Search

Database:	<input checked="" type="checkbox"/> US Pre-Grant Publication Full-Text Database <input checked="" type="checkbox"/> US Patents Full-Text Database <input type="checkbox"/> US OCR Full-Text Database <input type="checkbox"/> EPO Abstracts Database <input type="checkbox"/> JPO Abstracts Database <input type="checkbox"/> Derwent World Patents Index <input type="checkbox"/> IBM Technical Disclosure Bulletins
Term:	L1 and (ports with select\$ with (thread\$ or subroutine\$))
Display:	<input type="text" value="10"/> Documents in Display Format: <input checked="" type="checkbox"/> KWIC Starting with Number <input type="text" value="1"/>
Generate:	<input type="radio"/> Hit List <input checked="" type="radio"/> Hit Count <input type="radio"/> Side by Side <input type="radio"/> Image

Search History

DATE: Friday, April 30, 2004 [Printable Copy](#) [Create Case](#)

Set Name **Query**

side by side

DB=USPT; PLUR=YES; OP=ADJ

		Hit Count	Set Name
			result set
<u>L5</u>	L1 and (ports with select\$ with (thread\$ or subroutine\$))	7	<u>L5</u>
<u>L4</u>	L1 and (scan\$ with port\$ with select\$ with (thread\$ or subroutine\$))	1	<u>L4</u>
<u>L3</u>	L1 and (scan\$ with port\$ with select\$)	75	<u>L3</u>
<u>L2</u>	L1 and (scan\$ with port\$)	705	<u>L2</u>
<u>L1</u>	(712/\$.ccls. or 709/\$.ccls.)	23278	<u>L1</u>

END OF SEARCH HISTORY

First Hit Fwd Refs *Thread selec.*
L5: Entry 2 of 7

File: USPT

Aug 5, 2003

DOCUMENT-IDENTIFIER: US 6604125 B1

TITLE: Mechanism for enabling a thread unaware or non thread safe application to be executed safely in a multi-threaded environment

Detailed Description Text (23):

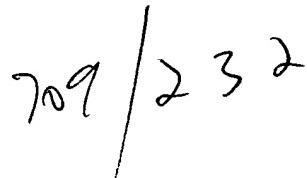
Once all of the thread pools are allocated, the server 106 is ready to service requests from clients 102. When a request is received (404) (at a communications port by a low level mechanism such as an operating system), a thread from the request processing mechanism thread pool is selected. This thread is assigned to and associated with the request, and is thereafter used to execute the computer code constituting the request processing mechanism 110 to process the request. Multiple requests can be processed concurrently; thus, if another request is received, then another thread is selected from the request processing mechanism thread pool, and that thread is used to execute the computer code constituting the request processing mechanism 110 to process the request.

Current US Cross Reference Classification (1):

709/219

Current US Cross Reference Classification (2):

709/226


709 / 232

h e b

b g e e e f

e b

e g



US00660412SB1

(12) **United States Patent**
Belkin

(10) Patent No.: **US 6,604,125 B1**
(45) Date of Patent: **Aug. 5, 2003**

(54) **MECHANISM FOR ENABLING A THREAD UNAWARE OR NON THREAD SAFE APPLICATION TO BE EXECUTED SAFELY IN A MULTI-THREADED ENVIRONMENT**

(75) Inventor: **Ruslan Belkin**, Mountain View, CA (US)

(73) Assignee: **Sun Microsystems, Inc.**, Palo Alto, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/574,314**

(22) Filed: **May 19, 2000**

Related U.S. Application Data

(60) Provisional application No. 60/156,305, filed on Sep. 24, 1999, and provisional application No. 60/155,711, filed on Sep. 24, 1999.

(51) Int. Cl.⁷ **G06F 9/00**

(52) U.S. Cl. **709/104; 709/100; 709/226; 709/219**

(58) Field of Search **709/100, 101, 709/102, 103, 105, 107, 108, 203, 206, 207, 219, 226, 315, 317, 318, 104**

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,410,698 A	4/1995	Danneels et al.	395/650
5,691,973 A	11/1997	Ramstrom et al.		
5,790,790 A	8/1998	Smith et al.		
5,793,415 A	8/1998	Gregory, III et al.		
5,796,393 A	8/1998	MacNaughton et al.		
5,809,145 A	9/1998	Slik et al.		
5,809,248 A	9/1998	Vidovic		
5,835,724 A	11/1998	Smith	395/200.57
5,872,963 A	2/1999	Bitar et al.		
5,918,228 A	6/1999	Rich et al.		
5,928,323 A	7/1999	Gosling et al.	709/203
5,961,584 A	10/1999	Wolf		
5,991,792 A	11/1999	Nageswaran		
6,012,090 A	1/2000	Chung et al.	709/219
6,052,711 A	4/2000	Gish		

6,112,196 A	8/2000	Zimowski et al.
6,182,109 B1	1/2001	Sharma et al.
6,374,286 B1	4/2002	Gee et al.
6,393,477 B1	5/2002	Paxhia et al.
6,418,458 B1 *	7/2002	Maresco
		709/103

OTHER PUBLICATIONS

,<http://mail.gnome.org/archives/orbit-list/2002-january/msg00152.html>*

MungeemSumedh, "Online COBRA Documents", Aug. 14, 1996, <<http://www.cs.wustl.edu/~schmidt/COBRA-docs/>>.*

"System for Dispatching from Multiple Thread Pools", Jan. 1, 1998, IBM Technical Disclosure Bulletin, vol. 41, issue No. 1, pp. 329-332.

* cited by examiner

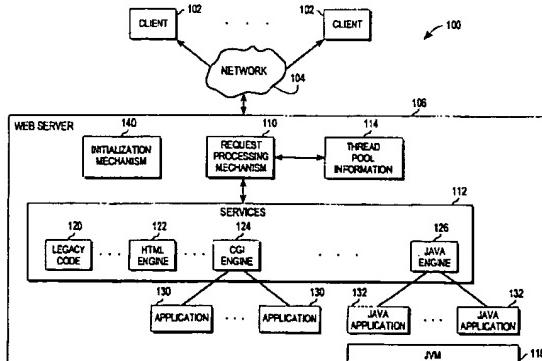
Primary Examiner—Majid A. Banankhah

(74) **Attorney, Agent, or Firm**—Bobby K. Truong; Hickman Palermo Truong & Becker LLP

(57) **ABSTRACT**

Executing a thread unaware or non-thread-safe application in a multi-threaded environment is potentially hazardous. If multiple instances of the thread unaware application are executed concurrently in the same process space, as may be the case in a multi-threaded environment, one instance may try to modify or overwrite the information used by the other instance, which can lead to serious errors. To enable a thread unaware application to be executed safely in a multi-threaded environment, multiple thread pools are implemented. That is, for each thread-unaware or non-thread-safe application, a separate thread pool is defined and associated with the application. Unlike other thread pools, though, this thread pool has its maximum number of threads parameter set to "1". By limiting the number of threads in the pool to 1, it is guaranteed that there will be no more than one instance of the thread unaware application executing at any one time. This in turn precludes any possibility of one instance of the application overwriting or modifying the information used by another. By implementing thread pools in this manner, execution of the thread-unaware application in a multi-threaded environment is made safe.

18 Claims, 6 Drawing Sheets



First Hit Fwd Refs Generate Collection

L5: Entry 7 of 7

File: USPT

Jul 7, 1998

DOCUMENT-IDENTIFIER: US 5778243 A
TITLE: Multi-threaded cell for a memory

Brief Summary Text (20):

It is an object of the present invention to achieve a multi-threaded memory having an optimally minimized number of transistors necessary to form read and write ports for the storage elements (corresponding to the threads of a multi-threaded cell) by recognizing that read independence and write independence are negligible functional advantages that can be selectively eliminated with negligible negative consequence, thereby improving the associated wireability which improves the surface area economy.

Detailed Description Text (5):

FIG. 3 depicts a block diagram of a first embodiment of the multi-threaded storage cell of the present invention. In particular, FIG. 3 shows a multi-threaded read interface including a thread selector 110 for selectively connecting threads 0 and 1 to the read ports 114. The number of read ports 114 corresponds to the number, usually more than one (e.g., six or eight), of functional units that can read from storage elements 104 and 108. An example of a functional unit is an arithmetic logic unit (ALU) that can perform integer, logical, shifting, field extraction and/or floating point operations and/or conversions between integer and floating point number representations. For a read operation, each storage element is connected to a functional unit as follows: the storage element is connected to the selector 110; the selector 110 is connected to one of the ports 114; this one port is connected to a decoder; and the decoder is connected to the functional unit.

Detailed Description Text (6):

FIG. 3 also includes write ports 102 for thread 0 and write ports 106 for thread 1 (the number of write ports corresponding in number to the number, usually more than one, e.g., three or twelve, of functional units that can write to the storage element 104), storage element 104 for thread 0 (connected to the write ports 102 and the thread selector 110), and storage element 108 for thread 1 (connected to the write ports 106 and the thread selector 110).

Detailed Description Text (7):

The multi-threaded cell of FIG. 3 reflects the inventors' discovery that consumption in chip-surface area can be reduced significantly, by elimination of read independence, without suffering substantial negative consequences because read independence is a negligible functional attribute for a single processor that can only access one thread at a time. Fewer transistors are required to form the thread selector 110 and the read ports 114 than would have been necessary to form separate read ports (as in the prior art) for each of thread 0 and 1, thereby achieving an increased surface area economy without suffering a significant negative consequence.

Detailed Description Text (8):

For the multi-threaded storage cell of FIG. 3, a processor (not shown) can read the data in the storage element 104 by controlling the thread selector 110 to select thread 0, thereby making the data of storage element 104 available on the read ports 114. Similarly, to read the data in the storage element 108, it is necessary

h e b b g e e e f

e b

e g

for the processor to control the thread select 110 to selector the line from the storage element 108.

Detailed Description Text (15):

FIG. 7 depicts a block diagram of a second embodiment of the multi-threaded storage cell of the present invention. FIG. 7 includes a single set of write ports 300 supplying data to a thread selector 302, a storage element 304 for thread 0 (connected to the thread selector 302), read ports 306 for thread 0 (connected to the storage element 304), a storage element 308 for thread 1 (connected to the thread selector 302), and read ports 310 for thread 1 (connected to the storage element 308). Once again, the number of write ports and read ports will correspond to the number of functional units that can write to and read from the storage elements 304 and 308, respectively. For a write operation, each functional unit is connected to a storage element as follows: the functional unit is connected to a write decoder; the write decoder is connected to one of the ports 300; this port is connected to the write selector 302; and the write selector 302 is connected to each of the storage elements.

Detailed Description Text (17):

FIG. 8 depicts a block diagram of a third embodiment of the multi-threaded storage cell of the present invention. In FIG. 8, the storage element 356 of thread 0 and the storage element 358 of thread 1 are both connected to a single set of write ports 352 via a thread selector 354. Similarly, the storage elements 356 and 358 are both connected to a single set of read ports 362 via a thread selector 360.

Detailed Description Text (18):

FIG. 9 depicts a block diagram of a content addressable version of the embodiment of FIG. 3. Elements that are the same as in FIG. 3 will be referred to by the same reference number. FIG. 9 differs from FIG. 3 only in that, not only the read port 114, but also the compare ports 402 receives the output of the thread selector 402. Content addressable memories are well known and only a brief amount of additional description will be provided.

Detailed Description Text (38):

The present embodiments have the advantage that additional threads can be added without the necessity of adding a separate set of write ports and read ports for each thread, due to the present use of thread selectors.

Current US Original Classification (1):

712/11

Current US Cross Reference Classification (1):

709/213

CLAIMS:

3. The multi-threaded memory of claim 2, wherein the read interface includes an N-selector and one port per said at least one read decoder.
4. The multi-threaded memory of claim 3, wherein the write interface includes an N-selector and one port per said at least one write decoder.
22. The multi-threaded memory of claim 21, wherein the read interface includes the N-selector and has at least one port per said at least one read decoder.



US005778243A

United States Patent [19]

Aipperspach et al.

[11] Patent Number: **5,778,243**[45] Date of Patent: **Jul. 7, 1998****[54] MULTI-THREADED CELL FOR A MEMORY**

[75] Inventors: Anthony Gus Aipperspach; Todd Alan Christensen; Binta Minesh Patel; Nghia Van Phan; Michael James Rohn; Salvatore Nicholas Storino; Bryan Joe Talik; Gregory John Uhlmann, all of Rochester, Minn.

[73] Assignee: International Business Machines Corporation, Armonk, N.Y.

[21] Appl. No.: **675,315**

[22] Filed: **Jul. 3, 1996**

[51] Int. Cl.⁶ **G06F 12/02; G06F 9/40**

[52] U.S. Cl. **395/800.11; 395/200.43;
395/823**

[58] Field of Search **395/377, 479,
395/706, 704, 872, 200.65, 441, 412, 435,
603, 200.43, 200.67, 200.57, 800.11, 800.28;
365/189.01, 230.01, 189.05, 230.08; 364/DIG. 1.**

DIG. 2

[56] References Cited**U.S. PATENT DOCUMENTS**

4,502,118	2/1985	Hageamaier, Jr. et al. 395/200.68
4,954,986	9/1990	Driessens et al. 365/189.01
5,051,947	9/1991	Messenger et al. 395/603
5,134,705	7/1992	Smith et al. 395/500
5,142,676	8/1992	Fried et al. 395/479
5,197,138	3/1993	Hobbs et al. 395/563
5,226,131	7/1993	Grafe et al. 395/377
5,237,662	8/1993	Green et al. 395/670
5,297,265	3/1994	Frank et al. 395/412
5,301,340	4/1994	Cook 395/800.24
5,331,643	7/1994	Smith 371/22.33
5,404,550	4/1995	Horst 364/232.22
5,430,850	7/1995	Papadopoulos et al. 395/683
5,448,709	9/1995	Chandler et al. 395/872
5,535,393	7/1996	Reeve et al. 395/706

FOREIGN PATENT DOCUMENTS

0614139A2 9/1994 European Pat. Off.
W09427216 11/1994 WIPO .

OTHER PUBLICATIONS

King et al., *IBM Technical Disclosure Bulletin*, vol. 19, No. 6, Reducing Complexity of Terminal Network Application Support Programming, (1976) pp. 2125-2129.

Paul H. Bardell et al., "Self-Testing of Multichip Logic Modules", 1982 IEEE Test Conference, pp. 200-204.

Hiroaki Hirata et al., "An Elementary Processor Architecture with Simultaneous Instruction Issuing from Multiple Threads", 1992 ACM, pp. 136-145.

Rafael H. Saavedra-Barrera et al., "Analysis of Multi-threaded Architectures for Parallel Computing", 1990 ACM, pp. 169-178.

R.S. Nikhil et al., "T: A Multithreaded Massively Parallel Architecture", 1992 ACM, pp. 156-167.

D.S. Renshaw, "Portable Co-Operative Multi-Threading Via Modified Procedural Cell", vol. 37, No. 10, Oct. 1994, pp. 283-284.

Kai Hwang, "Advanced Computer Architecture: Parallelism, Scalability, Programmability", McGraw-Hill, Inc., 1993, pp. 525-531.

Primary Examiner—Daniel H. Pan
Attorney, Agent, or Firm—Birch, Stewart, Kolasch & Birch

[57] ABSTRACT

A multi-threaded memory (and associated method) for use in a multi-threaded computer system in which plural threads are used with a single processor. The multi-threaded memory includes: multi-threaded storage cells; at least one write decoder supplying information to a selected multi-threaded storage cell; and at least one read decoder accessing information from a selected multi-threaded storage cell. Each of the multi-threaded storage cells includes: N storage elements, where $N \geq 2$, each of the N storage elements having a thread-correspondent content; a write interface supplying information to the intra-cell storage elements; and a read interface reading information from the intra-cell storage elements. At least one of the intra-cell read and write interfaces selects one of the thread-correspondent contents based at least in part by identifying the corresponding thread to achieve intra-cell thread-correspondent content selection.

35 Claims, 15 Drawing Sheets

